PROCEEDINGS, TOUGH Symposium 2012
Lawrence Berkeley National Laboratory, Berkeley, California, September 17-19, 2012

REDUCED ORDER MODELS FOR SUBSURFACE FLOW IN iTOUGH2

George Pau, Yingqi Zhang, Stefan Finsterle

Lawrence Berkeley National Laboratory
1 Cyclotron Road, Mail Stop 74-0120
Berkeley, CA 94720, USA
e-mail: gpau@lbl.gov, yqzhang@lbl.gov, safinsterle@lbl.gov

ABSTRACT

Inverse modeling involves repeated evaluations of the forward simulation, which can be computationally prohibitive for large numerical models. To reduce the overall computational burden of these simulations, we study the use of reduced order models (ROMs) as numerical surrogates. These ROMs usually involve using solutions at different sample points within the parameter space to construct an approximate solution at any point within the parameter space.

This paper examines a black-box relational approach based on Gaussian process regression. We demonstrate how an approximate error bound of the predicted solution can be constructed from the estimated variance of the approximation. We show that these ROMs perform better than look-up tables, particularly when the number of sample points is small. In particular, we show how these sample points can be chosen optimally to minimize computational efforts. Finally, we incorporate these ROMs within the inverse modeling framework of iTOUGH2 and demonstrate how ROMs can be used within that framework.

INTRODUCTION

The need to accurately simulate the multiscale dynamic behavior of multiphysics systems and inclusion of a variety of data has led to increasingly large and complex models in areas of geological CO_2 sequestration, nuclear waste disposal, environmental remediation, as well as the recovery of conventional (geothermal, oil, gas) and unconventional (hydrates, tight gas) energy resources. These simulations of nonisothermal flows of multicomponent, multiphase fluids in three-dimensional porous and fractured media may involve the iterative simultaneous solution of millions of coupled partial differential equations (PDEs) at each time step. While high-fidelity simulations are essential for understanding coupled processes, they may be computationally very expensive. As a result, it is impractical to use these models as the basis for conducting analyses that require many simulation runs (such as inverse modeling, parametric study of state variables, uncertainty analysis, and optimal design). High-fidelity models are needed to capture the physics of the problem with the required accuracy. The development of defensible reduced-order models for inversions and uncertainty quantification may offer a solution, but requires a careful evaluation of errors which we will use to inform our analysis.

Due to the complexity of subsurface simulation, most existing ROMs attempt to approximate the relationship between the parameters and outputs of interest using a response surface approach. In particular, lookup tables in combination with linear or higher-order polynomial interpolation are commonly used. However, polynomial interpolation is generally inaccurate (except for very smooth response surfaces) and not robust in the presence of uncertainties when the problem of interest is stochastic in nature.

In this paper, we consider the use of Gaussian process (GP) regression (Rasmussen and William, 2006) for ROM construction. It is a generalization of the kriging technique commonly used in geostatistics. We will briefly describe the GP regression model and how we can adaptively construct a ROM that minimizes the number of simulations needed to evaluate the outputs. We will then demonstrate its performance in several test problems, comparing it to an adaptive look-up table approach. We will finally describe an example in which this GP-based ROM is used within the iTOUGH2 framework.

METHODOLOGY

Gaussian Process Regression

Let us first give an abstract formulation for the response surface problem. Given a scalar function $f(\mathbf{p})$, where $\mathbf{p}=\{p_1, \ldots, p_n\}$ is a parameter vector of length n, we would like to approximate $f(\mathbf{p})$ by $g(\mathbf{p})$ using only known solutions of $f(\mathbf{p}\ p)$ for $\mathbf{p}$ in a sample set $\boldsymbol{S}_N=\{q_1, \ldots, q_N\}$ of size N.

A Gaussian process regression first assumes the relation between $\mathbf{p}$ and $f(\mathbf{p})$ can be described by a Gaussian process characterized by its mean function, $m(\mathbf{p})$, and covariance function, $k(\mathbf{p}, \mathbf{p}')$ (Rasmussen and William, 2006):

$$m(\mathbf{p}) = E[f(\mathbf{p})] \qquad (1)$$

$$k(\mathbf{p},\mathbf{p}') = E[(f(\mathbf{p}) - m(\mathbf{p}))(f(\mathbf{p}') - m(\mathbf{p}'))] \quad (2)$$

Knowing $f(\mathbf{q})$ where $\mathbf{q}\in\boldsymbol{S}_N$, the joint distribution of the $f(\mathbf{q})$ and $g(\mathbf{p})$ based on the above prior is then

$$\begin{bmatrix} f \\ g \end{bmatrix} : N\left(m(q), \begin{bmatrix} K(q,q) & K(q,p) \\ K(p,q) & K(p,p) \end{bmatrix} \right) \qquad (3)$$

where $K(\mathbf{q,p})$ is the covariance matrix. The joint posterior distribution of g(p) is then given by

$$\begin{aligned} &g(p)\,|\,q, f(q) \sim N(K(p,q)K(q,q)^{-1}f(q), \\ &K(p,p) - K(p,q)K(q,q)^{-1}K(p,q) \end{aligned} \qquad (4)$$

In other words, for any given p, the GP regression procedure gives the expected value and variance of the approximating function g(p):

$$E[g(p)] = K(p,q)K(q,q)^{-1}f(q) \qquad (5)$$

$$\sigma^2[g(p)] = K(p,p) - K(p,q)K(q,q)^{-1}K(p,q) \qquad (6)$$

The accuracy of the results then crucially depends on the priors for the mean and covariance functions over the entire parameter space. In this paper, we consider a constant mean function ($m(\mathbf{p}) = c_0$) and examine two different covariance functions.

The first covariance function we use is an isometric squared exponential (isoSE) function given by

$$K(\mathbf{p},\mathbf{p}') = \sigma_f \exp(-(|\mathbf{p}\text{-}\mathbf{p}'|/l)^2/2) + \sigma_n\delta_{\mathbf{p},\mathbf{p}'} \ (7).$$

The hyperparameters σ_f represent the variance at the point, l is the characteristic length, and σ_n represents the variance of the noise. This function depends on parameter distance $|\mathbf{p}\text{-}\mathbf{p}'|$ only, and is thus stationary. As a result, this covariance function is appropriate if f(p) varies smoothly in p. Without the noise variance σ_n, the covariance function is equivalent to an infinite linear combination of Gaussian radial basis functions. The resulting GP regression is then equivalent to a radial basis function interpolation. Note, however, that the GP regression is capable of modeling noise within the current formalism.

The second covariance function we considered is a neural network (NN) covariance function (Williams, 1998) given by

$$k(p,p') = \sigma_f \sin^{-1}\left(\frac{2pp'/l^2}{\sqrt{(1+2pp/l^2)(1+2p'p'/l^2)}} \right) + \sigma_n\delta_{p,p'} \qquad (8)$$

This is an inhomogeneous covariance that allows abrupt change in $f(\mathbf{p})$ that depends on the sign of $\mathbf{p}$. There are other covariance functions that one can use; a sample list can be found in Rasmussen and William (2006). The two covariance functions we have used here are, however, sufficient to illustrate the importance of choosing the appropriate priors.

In the above definitions, c_0, σ_f, and σ_n are known as the hyperparameters. To determine these hyperparameters, we will solve an optimization problem that maximizes the marginal Gaussian likelihood function, which is equivalent to minimizing the following negative log marginal likelihood (Rasmussen and William, 2006)

$$-\log(P(f|\mathbf{p})) = (1/2)(f^T K f + \log(|K|) + n\log(2\pi)) \quad (9)$$

For this work, we built our ROM based in part on Gaussian Process Regression and Classification (GPML) Toolbox version 3.1, but added additional functionalities, such as the adaptive sampling procedure. The optimization procedure used is based on the conjugate gradient method. Finally, we implemented the algorithms mentioned in this paper within iTOUGH2, allowing us to use the resulting reduced-order model for uncertainty quantification.

Sampling procedure

Eq. (5) shows that results will largely depend on the samples selected. The question remains, how can we select parameters in S_N? One may use statistical approaches, such as the Latin Hypercube sampling procedure, to determine S_N, but we will still need to determine the number of samples needed. Here, we examine an adaptive approach known as the "greedy algorithm," or (in the context of neural networks) the forward selection algorithm. The greedy algorithm has been proposed in Carr et al. (2001) for radial basis functions and in many other references (e.g., Rozza, 2007) for other ROM approaches.

In the greedy algorithm, we first construct a large search sample set S_S that sufficiently represents the entire parameter space. Starting with a randomly selected parameter point $\mathbf{p}_1 \in S_S$ we compute $f(\mathbf{p}_1)$ and construct our first ROM, $g_1(\mathbf{p})$ based on $S_1=\{\mathbf{p}_1\}$. We then determine $\mathbf{p}_2^* = \arg\max_{\mathbf{p}\in S_S} \mathrm{e}_1(\mathbf{p})$ where e_1 is an appropriate error measure of $g(\mathbf{p})$ -$f(\mathbf{p})$. Then, we append $\mathbf{p}_2$* to S_1 to form S_2. We repeat the procedure to construct ROM $g_2(\mathbf{p})$, …, $g_n(\mathbf{p})$ until either $\mathrm{e}_{\max}$= $\mathrm{e}_n(\mathbf{p}_{n+1}*)$ is below a predetermined error tolerance, or the number of sample points in S_S reaches the maximum allowable number.

In this work, we use σ (from Eq. 6) as our error measure e_n. Thus, we do not need to evaluate $f(\mathbf{p})$, $\forall \mathbf{p} \in S_S$. If our final ROM consists of N sample points, we only perform N full simulations. We will examine how this affects the distribution of points in S_N. We may use other error measures, but the choice will affect the accuracy and efficiency of the ROM. For example, one may choose to use the actual absolute error. Although this may lead to a more accurate model, it requires $f(\mathbf{p})$ to be predetermined for $\forall \mathbf{p} \in S_S$. We intend to examine other error measures that may lead to more accurate and efficient ROM in the future.

The above construction procedure also leads to a series of hierarchical ROMs that are increasingly more accurate. At each iteration n, we build a ROM that perform optimally given S_n and $f(\mathbf{p})$, $\mathbf{p} \in S_n$. For a GP regression model, we thus optimize the hyperparameters in every iteration. We then have at our disposal a series of ROMs that we can use, depending on the accuracy and efficiency needed in our application.

RESULTS

Sample problem

This test problem is based on the iTOUGH2 sample problem 6 (Finsterle, 2007), in which the forward model describes a ventilation experiment conducted at the Grimsel Rock Laboratory, Switzerland. The purpose of this particular test is to quantify the extent of the two-phase region and to study its hydraulic properties. *In situ* measurements of water potential, water content, temperature, and ambient air humidity were performed. Details of the forward model can be found in Finsterle and Pruess (1995).

The three uncertain parameters considered in the inverse problem are the logarithm of the absolute permeability, log(k), and the van Genuchten parameters n and log(1/α). For our purpose, we select capillary pressure at one point location only as the model output of interest, f(p).

The ranges of log(k), n and log(1/α) that we have considered are [-19,-14], [2,3] and [5,6]. Figure 1 shows how f(p) varies with log(k) and n for selected values of log(1/α). There is clearly a sudden change between log(k) = -16 and -14 but for log(k) between [-19,-16], f(p) appears reasonably smooth.

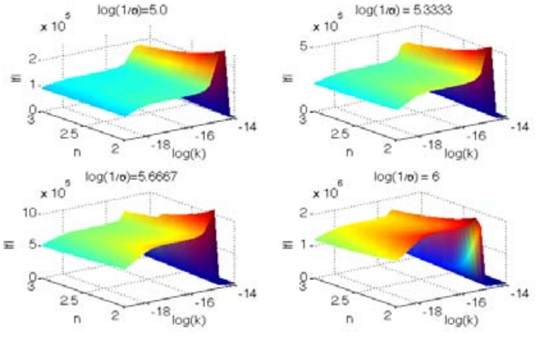


Figure 1. Model output $f(\mathbf{p})$ as a function of three input parameters for the sample problem.

We first consider two cases in which log(k) stays within a smaller range, [-19,-16], and a larger range, [-19,-14]. Ranges for n and log(1/α) are the same for both cases as given earlier. In a second analysis, we will use the constructed ROM to do an uncertainty quantification analysis in iTOUGH2. Note that we normalize the parameters such that they vary between 0 and 1. We set N to be 30. The search sample set, S_S, which we use for our greedy algorithm is 22 points in each direction.

Sample space

We first examine how the parameter points selected from the adaptive procedure are distributed in the sample space. With the isoSE covariance function, the distribution of sample points for the two cases is shown in Figure 2. Since the adaptive algorithm attempts to minimize the variance of the approximation (Eq. 6), which mostly depends on the distance between two points (Eq. 7), the selected samples are distributed almost but not exactly uniformly across the domain, independent of the behavior of f(p).

Based on the above observation, we constructed a ROM based on a uniform distribution of 27 points in the parameter space (each direction is uniformly divided into two intervals). The resulting maximum, mean and standard deviation of the errors are 0.056, 0.019 and 0.013. If we set N=27 and allow the adaptive algorithm to determine the points, these quantities are 0.062, 0.021 and 0.015. The slightly poorer performance is probably due to the initial poor approximation resulting from the small number of parameter points used to construct the ROM, leading to poor initial selection of the points.

Note that the apparent poorer performance of the adaptive algorithm should be put into context. With just 3 additional points, we are able to reach the same performance as uniform grid, as indicated by Table 1. This is obtained without the insights that we concluded from the previous paragraph. Indeed, the adaptive algorithm will work with any error measure for which insights on the optimal layout of sample points are not readily available or obvious. Finally, it is not necessary to start the adaptive algorithm with 1 parameter point. We can then easily start the adaptive algorithm with a larger S_n that incorporates these insights.

Approximation

To quantify the actual error, we use the relative error of the approximation:

$$e_{\mathrm{rel}}(p) = \frac{|f(p) - g(p)|}{|f(p)|} \qquad (10)$$

for p within a test sample set (we use the search sample set, S_S), and determine maximum, mean and standard deviation of $e_{\mathrm{rel}}(\mathbf{p})$. To evaluate $e_{\mathrm{rel}}(\mathbf{p})$, we thus need to evaluate $f(\mathbf{p})$ for all $\mathbf{p}$ in S_S.

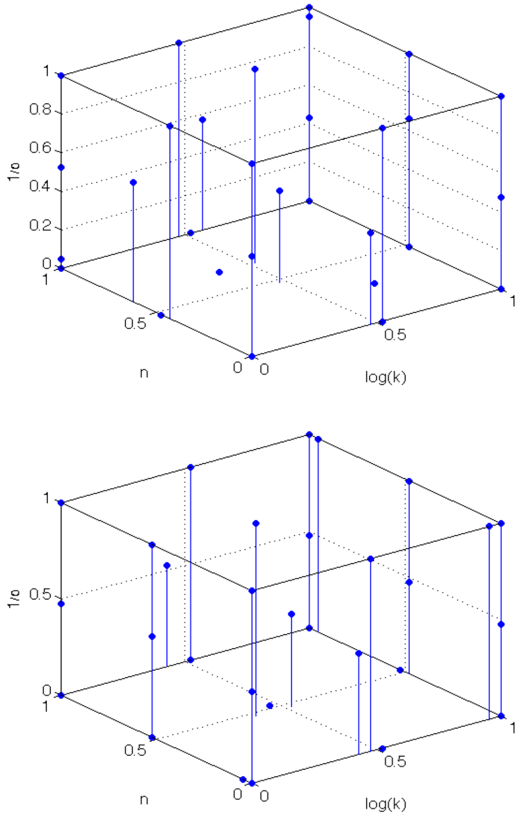


Figure 2. Distribution of sample points for smaller log(k) range (top) and longer log(k) range (bottom).

Table 1 shows the results for two ranges using the isoSE covariance function and the results for the larger range of log(k) using NN. Table 2 shows the results for two ranges using linear interpolation.

Table 1. Maximum, mean and standard deviation of approximation errors of GP regression with N=30.

Range, covariance function	$e_{\mathrm{rel}}(p)$		
	maximum	mean	Standard deviation
[-19,-16], isoSE	0.057	0.017	0.012
[-19,-14], isoSE	1.60	0.31	0.26
[-19,-14], NN	2.40	0.26	0.32

Since f(p) is a smooth function for the smaller range of log(k), Table 1 shows that the isoSE covariance function was able to approximate f(p) accurately. The maximum relative error is only 5.7%, and the mean error is 1.7% with only 30 samples. Compared to a linear interpolation procedure, the performance is significantly better for the same number of points.

Table 2. Maximum, mean and standard deviation of approximation errors of linear interpolation, N= 30.

Range	$e_{\mathrm{rel}}(p)$		
	maximum	mean	Standard deviation
[-19,-16]	0.096	0.017	0.018
[-19,-14]	4.42	0.28	0.43

For the larger range of [-19,-14], the accuracy of the approximation deteriorates, especially in the region where there is a large jump in the solutions shown in Figure 1. The isoSE covariance function is thus not an appropriate covariance function to use.

The NN covariance function is inhomogeneous and is expected to model the jump more accurately. However, based on Table 1, the resulting errors appear to be comparable to those obtained using the isoSE covariance function. This is because the inhomogeneity being modeled by NN covariance function is incompatible with our data; NN covariance function is suited for data that have abrupt change when **p** changes from positive to negative. However, 50% of the sample points in S_S have errors below 10% when the NN covariance function is used compared to 20% when the isoSE covariance function is used. The optimized hyperparameters of the NN covariance function are l = 0.072 and σ_f = 2.47. This short characteristic length and large variance reflects the attempt of the NN covariance function to model the jump. With the isoSE covariance function, the hyperparameters are l = 4.20e-1 and σ_f = 1.83. Here, the jump is not sufficiently captured since most of $f(\mathbf{p})$ is smooth in the parameter space.

It is clear that both covariance functions do not provide the accuracy we need. A more appropriate covariance function is one where the hyperparameters are function of **p** (Plagemann, 2008). However, the expected optimization problem will be arduous, and the resulting covariance function is harder to interpret.

In all of the above approximations, the hyperparameter σ_n is close to zero, because we are approximating the solution obtained through a deterministic simulation. However, the presence of σ_n implies that we could model noise in our solution. This will be explored in the future in the context of flow through heterogeneous formations.

One should note that these hyperparameters are obtained through a local optimization algorithm (steepest descent), and are thus sensitive to the starting position. This could also explain why the previous errors from both the isoSE covariance function and NN are not satisfying. More optimal hyperparameters may be obtained if global optimization algorithm is used, leading to better ROMs. The use of global optimization algorithm will be explored further in the future.

Uncertainty quantification

The purpose of developing such a ROM is to substitute a time-consuming high-fidelity model by a ROM in an inverse analysis or sampling-based uncertainty quantification (UQ) analysis, where many forward model evaluations are needed. We have implemented such capability into iTOUGH2, and performed UQ for the same sample problem (we have considered the smaller range of log(k)).

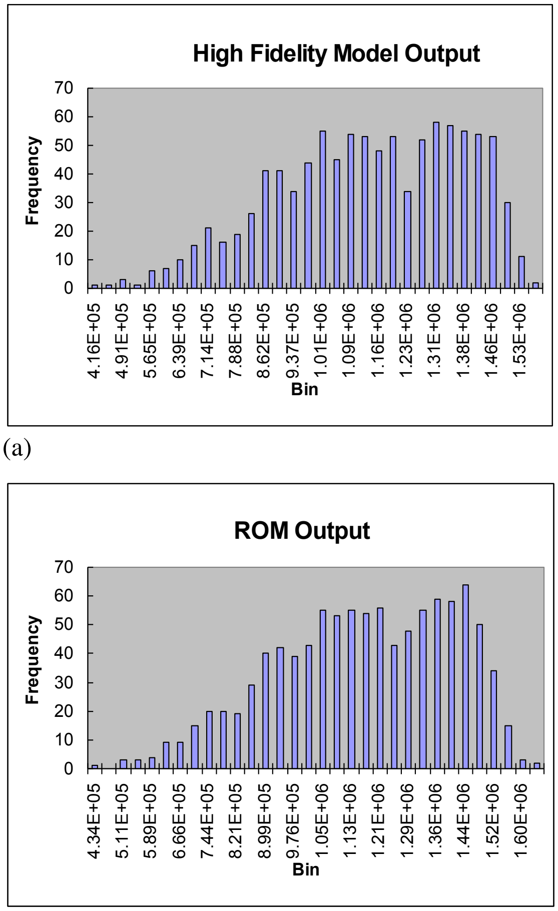
For comparison purposes, we performed an UQ using the high-fidelity model (HFM). The Monte Carlo simulation is performed with a sampling size of 100 and 1000 for both the ROM and HFM. The mean and variance of the model output from each Monte Carlo simulation are listed in Table 3.

Table 3. Comparison of UQ results between a HFM and its corresponding ROM

	HFM-1000	ROM-1000	HFM-100	ROM-100
Mean	1.11e6	1.15e6	1.14e6	1.17e6
Standard deviation	2.4e5	2.4e5	2.3e5	2.3e5

The ROM seems to be able to reproduce the standard deviation of the UQ analysis. The error of the mean estimation using the ROM is about 3%. For this particular example, 100 samples seem to be sufficient for uncertainty quantification.

The histogram using 1000 samples are plotted in Figures 3 (a) and (b). The ROM appears to reproduce the histogram of the model output relatively well. In this particular problem, it does not seem necessary to have a large number of samples for an UQ analysis (i.e., 100 samples is sufficient). However, most problems, especially the nonlinear ones with many sensitive and uncertain parameters, may need many forward evaluations, in which case ROM would save time. The ROM construction in this example requires only 30 forward HFM evaluations. Since the evaluation of the ROM during the Monte Carlo sampling has a negligible computational cost compared to running a single simulation with the HFM, the cost savings are proportional to $N_{MC}/30$, where N_{MC} is the number of Monte Carlo simulations.



(a)

(b)

Figure 3. Histogram of the Model output from the Monte Carlo simulation with 1000 samples, using both (a) HFM and (b) ROM

CONCLUSION

In this work, we examined two types of ROMs to approximate a high-fidelity model for inverse analysis: A Gaussian Process (GP) regression model and neural networks. We provided an error-estimation method for the proposed ROM methods. We improved the performance of GP-based ROM by implementing an adaptive sampling approach, so the ROM can be constructed with a minimum amount of expensive HFM simulations. In the sample problem, the ROMs using both approaches perform significantly better than a linear interpolation approach. However, the performance is not very satisfying when the model output experiences sudden changes. This implies that our prior models for the mean and covariance are inappropriate.

Constructing the ROM requires some CPU time, specifically the estimation of hyperparameters, which in itself is an optimization problem. However, this computational cost is relatively small and just a one-time effort. Once a ROM is constructed, the computational savings are demonstrated by a UQ analysis. The savings can be large for a big problem (i.e., problems with many uncertain parameters, large sample size, and each HFM evaluation taking a long time).

ACKNOWLEDGMENT

This work was conducted as part of Berkeley Lab's National Risk Assessment Partnership (NRAP) effort, supported by the Assistant Secretary for Fossil Energy, Office of Sequestration, Hydrogen, and Clean Coal Fuels, of the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231 and TOUGH2 development grant.

REFERENCES

Carr J.C., R.K. Beatson, J.B. Cherrie, T.J. Mitchell and T.R. Evans, Reconstruction and Representation of 3D Objects with Radial Basis Functions, *ACM SIGGRAPH 2001*, 12-17 August 2001, Los Angeles, CA, USA.

Finsterle, S., and K. Pruess, Solving the estimation-identification problem in two-phase flow modeling, *Water Resour. Res.*, *31* (4), 913–924, 1995.

Finsterle, S., *iTOUGH2 Sample Problems*, Report LBNL-40042, Lawrence Berkeley National Laboratory, Berkeley, Calif., 2007.

Plagemann C., K. Kersting and W. Burgard, Nonstationary Gaussian Process Regression Using Point Estimates of Local Smoothness, In *Machine Learning and Knowledge Discovery in Database, Lecture Notes in Computer Science*, 5212, 204-219, 2008.

Rasmussen, C.E. and C.K.I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

Rozza, G., D.B.P. Huynh and A.T. Patera, Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations: Application to Transport and Continuum Mechanics, *Archives of Computational Methods in Engineering*, 15(3), 229–275, 2007.

Williams, C.K.I., Computation with Infinite Neural Networks, *Neural Computation*, 10(5), 1203–1216, 1998.